异构环境下纠删码的数据修复方法综述 *

钟凤艳，王 艳，李念爽

(华东交通大学 软件学院，南昌 330013)

摘 要：在大规模云存储系统中，由于磁盘或网络故障造成的存储节点失效事件频发，系统需要数据冗余技术以保证数据的可靠性和可用性。目前针对纠删码的冗余数据修复研究大都无差别地对待每个存储节点，然而实际分布式存储系统中，节点通常存在带宽资源、计算资源、存储容量资源等方面的差异性，这些资源的异构性对冗余数据修复性能影响很大。指出影响修复性能的关键因素，选取带宽开销、磁盘访问开销、修复时间、参与修复的节点数量和修复代价作为修复性能的评价标准；分析了现有研究方法如何降低这五种开销，重点讨论了这些方法的优缺点；阐述当前异构分布式存储系统中纠删码修复技术的研究现状；最后指出纠删码数据修复技术中尚未解决的一些难题和未来纠删码修复技术可能的发展方向。

关键词：存储系统；纠删码；异构；数据修复；性能优化

中图分类号：TP309.3 doi: 10.3969/j.issn.1001-3695.2018.04.0269

survey of heterogeneous-based data repair strategies for erasure codes

Zhong Fengyan, Wang Yan, Li Nianshuang

(*School of Software*, *East China Jiaotong University*, *Nanchang Jiangxi* 330013, *China*)

Abstract: In large scale cloud storage system, storage nodes fail frequently because of disk/link failures. In order to preserve data's reliability and availability, storage systems need to store redundant data. Compared with data-replication, erasure code can provide significant savings in storage space at the cost of extra data recovery overhead. Most of the data recovery mechanisms for erasure codes think that there is no difference among storage nodes. In the actual distributed storage systems, however, there are some heterogeneities in the bandwidth, computing resources and storage capacity of the storage nodes, which have a great influence on the redundant data recovery performance. This paper presents data repair operation process in erasure code technology under heterogeneous environment, and identifies the key factors that affect the recovery performance. Thereafter, this paper chooses bandwidth cost, disk access overhead, repair time, the number of nodes participating in the repair and repair cost as the evaluation criterion for the recovery performance. In addition, this paper analyzes how to reduce these five costs by the existing research methods, and focuses on the advantages and disadvantages of these methods. Finally, this paper indicates Some unsolved problems in the research of erasure code repair method in the current heterogeneous storage systems and the possible development direction of the future erasure code repair method.

Key words: storage systems; erasure code; heterogeneity; data recovery; performance improvement

0 引言

随着存储系统中存储节点数目的增加，以及存储节点的多样化和复杂化，节点常常发生失效[1]。为了对抗因节点失效而造成的数据丢失，分布式存储系统需要保存一定量的冗余数据来保证系统中存储数据的可靠性和可用性。产生冗余数据的技术有副本和纠删码两种。副本技术是把数据复制多个副本分别存储起来，当某个副本所在的存储节点出现故障时，分布式存储系统能够自动将服务切换到其他的副本，从而实现数据的高可靠和高可用。这种方法不涉及专门的编码和重构算法，容错性能较好，但存储利用率极低。随着数据的持续增长，在 PB 级别的数据中心，多副本技术会引入极大的存储开销。比如，现有的分布式存储系统，如 HDFS（Hadoop distributed file system）[2]、Ceph[3]，通常采用三副本的方式，这将消耗 3 倍原文件大小的存储空间。

面对这种情况，纠删码技术因具有存储开销低的优势被广泛应用。其中，RS 编码（Reed-Solomon code）[4]是目前,广泛使用的纠删码方案，其编码过程是将 k 个数据块按照一定的编

收稿日期：2018-04-17；**修回日期：**2018-06-12 **基金项目：**国家自然科学基金资助项目（61402172）；江西省教育厅资助项目（GJJ150509）

作者简介：钟凤艳（1993-），女，江西赣州人，硕士，主要研究方向为分布式存储（1215396009@qq.com）；王艳（1982-），女，副教授，博士，主要研究方向为分布式存储、数据可靠性等；李念爽（1993-），男，硕士，主要研究方向为分布式存储.

码规则生成 m 个校验块，对于这 $k+m$ 个编码块，其编码性质保证通过任意的 k 个编码块均能重建原始文件。以 RS(4,2)编码为例，将原文件分为 k=2 部分，然后按照 RS 码编码规则生成 m=2 个校验块，容错能力为 2，数据收集节点可以选择任意 2 个节点重建原始文件，此编码方式仅需消耗 1.5 倍原文件大小的存储空间，就具有与三副本技术相同的容错能力。

纠删码技术虽然具有存储开销低的优势，但存在修复成本较高的缺陷。当某个节点失效时，为了保持系统的冗余度，系统需要修复失效节点上的数据块并将其放置在其他正常的节点上。修复一个数据块时，副本方式通过其他正常节点拷贝相应的数据完成数据修复，纠删码方式需要从其他多个供应节点分别下载一个数据块才能修复出丢失的数据块。这导致修复时网络带宽开销较大和时间较长。在大规模集群环境下，磁盘、服务器和网络的错误已经成为常态。Rashmi 等人[5]监测了 Facebook 数据中心的一个使用(10,4)RS 纠删码的大集群的节点故障情况。被监测的集群里包含了超过 3000 个节点，每个节点上存储 15 TB 的数据，总共存储的数据量有 30 PB。根据他们的监测结果，在这样一个集群里，平均每天失效的存储节点数量就超过 20 个，而一天内失效的数量甚至可高达 100 个。在这种情况下，为了保证系统的高可靠和高可用，存储系统需要频繁进行修复操作，这加重了系统的压力。

针对纠删码修复数据存在的性能缺陷，近年来做了许多同构场景下（无差别对待每个存储节点）的理论设计和工程实现的工作。执行修复操作时无差别地对待每个存储节点，把每个存储节点的带宽资源、 计算能力资源和存储能力资源看成是一致的。目前，国内外存在少量有关纠删码数据修复技术的研究综述[6~10]，文献[8]主要介绍当前典型和常见的纠删码技术的发展现状，文献[9]主要关注编码方案、数据修复和数据更新等方面的最新研究进展，文献[10]主要围绕同构环境下计算、读写、传输三方面对优化纠删码修复性能的关键技术进行了探讨。本文将只聚焦于纠删码在异构分布式存储系统中的数据修复性能优化问题，我们认为异构分布式存储系统对纠删码数据修复性能的影响因素主要有以下三方面：存储节点间的带宽异构、存储节点计算能力异构以及存储节点的存储容量异构，因此本文将异构环境下的数据修复方法归为三类，包括面向带宽异构、面向计算能力异构和面向存储能力异构的数据修复方法。围绕修复带宽开销、磁盘访问开销、修复时间、参与修复的节点数量和修复代价 5 个指标，讨论现有的纠删码数据修复方法在这五个指标上的表现。最后指出纠删码数据修复方法未来可能的研究方向。

1　基本概念与影响因素

1.1　基本概念

为了便于理解，往往将一个服务器称为一个节点，下面对本文出现的相关概念给出如下说明。

a）数据块。原始用户数据被系统划分形成的最小编码单元。

b）校验块。原始数据块经过编码运算得到的结果。

c）条带。多个数据块与其对应的校验块构成的冗余集合，如果一定数目的编码块丢失，可以通过对所在条带中剩余编码块进行运算而重新生成。

d）MDS 码。MDS 码是使用空间最优的编码。将原文件切分为 k 块，(n,k)MDS 码可以将这些块编码为 n 个编码块，每个编码块的大小为原文件的 $\frac{1}{k}$，且其中任意 k 个编码块均可以重构出原文件，该性质称为 MDS 性质。满足 MDS 性质的编码均可以称为 MDS 码。

e）供应节点。参与数据修复的节点。供应节点读取本地数据后通过网络传输给其他节点来参与数据重建。

f）新生节点。重建丢失数据的节点。它需要通过从供应节点中收集所需数据，计算得到丢失的数据。

g）瓶颈带宽。修复拓扑中带宽最小的链路，瓶颈带宽决定了修复过程所需的时间。

1.2　修复性能的影响因素

数据修复过程一般包含四个步骤：

a）供应节点从本地磁盘读取所需数据。

b）为了减少数据传输量，供应节点对数据进行本地随机线性组合后生成传输数据。

c）供应节点将传输数据通过网络传输到新生节点。

d）新生节点接收到所有供应节点传来的数据后，解码恢复出丢失数据。

通过分析数据修复过程，在异构分布式存储系统中，对纠删码数据修复性能存在影响的因素主要有以下三方面：存储节点间的带宽异构、存储节点计算能力异构以及存储节点的存储容量异构。数据修复时间受到节点的读写能力、计算能力、传输能力的共同影响。由于大部分分布式系统都搭建在廉价的服务器上，这些服务器节点之间通过网络互联。然而，服务器节点是不可靠的，网络也是不可靠的。因此，服务器节点的 CPU 计算能力、 磁盘性能和网卡速度均有可能成为制约修复时间的瓶颈。

基于以上分析，我们认为存储节点间的带宽异构、计算能力异构和存储容量异构是影响纠删码修复性能的主要因素。因此，从这三方面入手， 将现有的数据修复方法归纳为面向带宽异构的修复方法、面向计算能力异构的修复方法和面向存储容量异构的修复方法，并选取修复带宽开销（repair bandwidth），磁盘访问开销（disk I/O），参与修复的节点数量（repair degree），修复时间和修复代价作为纠删码修复性能的评价标准。

2　面向带宽异构的修复方法

带宽异构是指存储节点间的链路带宽不总是相等，文献[11]指出实际的分布式存储系统中存储节点间的链路带宽存在差异性，有的链路带宽高，有的链路带宽低。由于供应节点给新生

节点传递的数据量都相等，当每个供应节点和新生节点之间链路带宽不相同时，完成一轮数据修复操作的时间就由带宽最小的链路决定（节点间的数据传输可以并行），因此一个很自然的想法就是根据链路带宽动态地调整每个供应节点传输给新生节点的数据量，使得带宽大的链路传输较多数据，带宽小的链路传输较少数据。现有的研究工作也都是按照这个思路来优化数据修复性能。本节介绍三类带宽异构下的数据修复策略，包括星型拓扑下的弹性修复策略、树型拓扑下的修复策略和基于XOR 的纠删码修复技术，并分析这些方法在修复带宽开销、修复时间、参与修复的节点数量和修复代价上的表现。

2.1　星型拓扑下的弹性修复方法

基于星型结构的串行修复策略(Star Structure Based,SSR)[12]是指当多个节点同时失效后，系统会按照串行的方式依次修复失效节点，重构多个冗余数据节点，恢复原有冗余度。在构建每个冗余数据节点时，系统会构建以新生节点为中心、提供节点为边界的星型结构，所有供应节点直接向新生节点传输数据。在此结构中，再生时间是由新生节点与供应节点之间最慢的一条带宽链路决定，下面介绍 4 种基于星形修复模型的修复方法。

2.1.1 最大弹性选择的弹性修复

弹性修复是指在每一轮修复中，根据可用带宽大小动态地决定相应的数据传输量，具体来说就是带宽大的链路传输更多数据量，反之，则传输较少数据量。Dimakis 等人[13,14]介绍的再生码数据修复过程存在一个较强的限制：每个供应节点向新生节点传输等量数据进行数据修复，数据收集节点(DC 节点)仅连接 k 个节点，并从每个节点下载 $\frac{M}{k}$ 数据量。这样做的话，那些可以使用更多链路的新生节点或 DC 节点也只能使用一部分链路，这样可能会浪费一些带宽高的链路资源，从而可能导致修复效率不高。Shah 等人[15]提出弹性修复策略，允许供应节点和 DC 节点充分利用可用链路资源向新生节点传输不等量数据。具体来说，DC 节点从节点 $i(1\leqslant i\leqslant n)$ 下载 $\mu_i(0\leqslant\mu_i\leqslant\alpha)$，满足总下载量不小于 M（原文件的大小）即可，同样，新生节点从供应节点 $i(1\leqslant i\leqslant n)$ 接收 $\beta_i\left(0\leqslant\beta_i\leqslant\beta_{\max}\right)$ 数据量,满足总接收量大于或等于一个设定参数 γ 即可，可表示为下面两个不等式：

$$\sum_{i=1}^{n}\mu_i \geq M, 0\leq \mu_i \leq \alpha \tag{1}$$

$$\sum_{i=1}^{n}\beta_i \geq \gamma, 0\leq \beta_i \leq \alpha \tag{2}$$

当一个节点失效时，新生节点选择任意 k 个节点下载 $\frac{M}{k}$ 数据量，修复带宽等于原文件的大小。此修复模式可能没有利用到一些带宽较高的链路，如果用弹性策略，新生节点可以根据可用带宽的大小从不同节点上下载不等量数据来降低再生时间。

2.1.2 最优节点选择的修复方法

现有的文献中关于降低再生时间的方法通常是两种，一种是降低数据传输量，另外一种是变换再生过程的拓扑结构，然而文献[16]认为不同的供应节点参与修复数据过程也会影响修复性能，因此对两种情况：a）供应节点确定，新生节点不确定；b）供应节点和新生节点都不确定，分别提出了 FPSN 和SPSN 算法选择最优供应节点。FPSN 算法思想是固定 d 个供应节点，选择一个最优新生节点，从而形成一个瓶颈带宽最大的修复拓扑，SPSN 算法思想是遍历所有链路带宽，在 d 个供应节点和一个新生节点组成的所有可能的修复拓扑结构中寻找最小链路带宽最大的修复拓扑结构。另外，文献对第二种情况设计了 FLEX 算法来计算每个供应节点传输给新生节点的数据量。实践证明，使用文献提出的节点选择方案，平均修复时间可以减少 58.56%。

2.1.3 分布式存储系统中的下载成本与修复带宽的权衡

由于分布式存储系统中不同类型的存储节点存在带宽异构的差异性，修复失效数据时，新生节点选择不同供应节点下载一个数据块的成本开销也可能不一样，即下载成本异构，Akhlaghi 等人[17] 假设系统中存在两种类型的下载成本 C_1 和 C_2，相同下载成本类型的节点组成一组，不同组的节点下载成本不同。文献利用信息流图，提出了广义再生码 (GRC)，从理论上比较了广义最小存储再生码(GMSR)和最小存储再生码(MSR)以及广义最小带宽再生码(GMBR)和最小带宽再生码(MBR)的修复成本和下载成本，在某些特定情况下 GRC 码优于 RC，文献仅解决了下载成本和修复带宽权衡关系的问题，没有给出下载成本异构情况下的数据修复方案。

2.1.4 面向纠删码的低成本多节点失效修复方法

前面介绍的基于星形修复方法都仅讨论了单个节点的失效修复问题，对于多节点失效修复问题没有提到。在实际系统中经常会发生多节点失效的情况[18,19],郑等人[20]提出了面向纠删码的低成本多节点失效修复方法，采用串行的修复方式依次完成多个失效节点的修复工作，把网络距离作为节点选择的依据。他们认为网络距离较短的节点之间拥有更高的带宽，反之，则拥有更小的带宽。具体修复过程如下，假设 r 个节点失效，在进行修复操作时，先选定 r 个新生节点，然后从 n-r 个节点中选择 k 个节点，这 k 个节点需满足其到 r 个新生节点的总网络距离最短，确定供应节点后，从这些新生节点中选则其中的一个节点作为中心节点，该节点同时与其他新生节点和供应节点进行通信。在确定中心节点后，中心节点从 k 个供应节点分别接收 $\frac{M}{k}$ 数据量,供应节点仅需传输一次数据，中心节点就可以完成 r 个失效块的构建工作，最后将其中对应的一个数据块存储在本地，并将其余的 r-1 个数据块分别发送到其他 r-1 个新生节点。从此修复过程可以分析出，中心节点的选择需要同时考虑其与供应节点和其他新生节点之间的网络距离。

这种通过基于网络距离来选择供应节点和中心节点的方法可以提高节点之间的可用带宽，给系统减少了测量节点之间可用带宽的负担。另外，采用多线程的计算方式和流水线的数据传输方式组织数据的计算和传输，并使用基于中心节点的数据修复方式同时修复多个失效数据块，极大地减少了带宽开销。

2.2 树型拓扑下的修复方法

星形结构下的修复方式较为简单，缺点是中心节点同时承受计算任务和传输任务。为了提高数据传输效率，李钧等人[21,22]设计了一个新的传输拓扑--树形拓扑，用以替换当前修复过程中使用的星形拓扑，从而达到提高数据修复过程中数据传输速度的目的。为了最大化利用网络链路资源，他们将树型拓扑的链路选取建模为图论中的瓶颈生成树问题，并提出了相应的最优算法解决该问题。他们首先考虑的是供应节点数量为 k 的情况，即典型的 MDS 码。随后，他们继续考虑了 MSR 码的树形修复问题，以及建立多棵树并行来解决双向链路带宽不同的情况。下面用三个小节分别介绍李钧等人的工作，并讨论他们工作的优缺点。

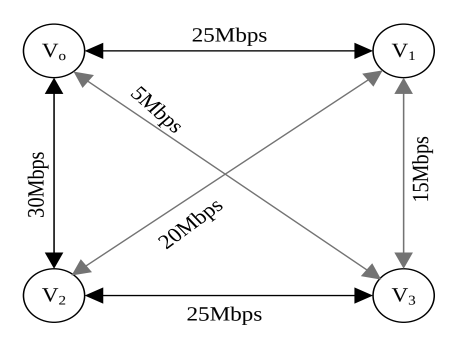
2.2.1 对称链路单颗树的再生过程

在对称网络中，节点之间上行和下行的带宽相等称为对称链路。为了提高瓶颈带宽，李钧等人构造了对称链路单棵树的再生过程，使用 Prim 算法构造最优再生树，每条链路传输 $\frac{M}{k}$ 数据量，并且允许中间节点提前对数据块进行编码。通过建立树型传输路径提高瓶颈带宽，达到降低修复时间开销的目的。

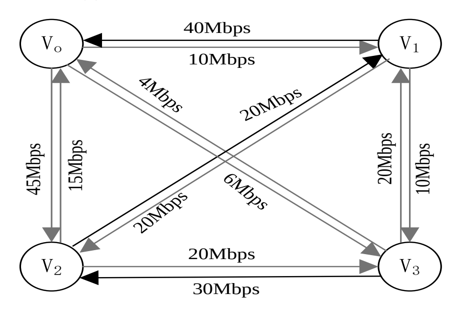
李钧等人提出的树形修复方法，可以很好的适应实际分布式存储系统中可用带宽不一致的场景，大大节省了存储系统的修复时间。然而，王艳等人[23]通过实验发现，李钧等人提出的针对 MSR 编码的树型修复策略,在修复时不能很好的保证数据完整性。因为在修复过程中同时改变修复拓扑结构并允许中间节点进行编码的情况下，会导致修复操作中所传递的信息量不够修复出所丢失的数据。于是，王艳等人通过分析树形修复的信息流图，得到了修复操作中每条边上所需传输的信息量最小值。他们不仅很好的弥补了李钧等人在树形 MSR 编码上信息量不足的问题，而且还提出了一种新的同样适用于带上宽异构的分布式存储系统中的数据修复策略--弹性修复策略。

2.2.2 非称链路单棵树的再生过程

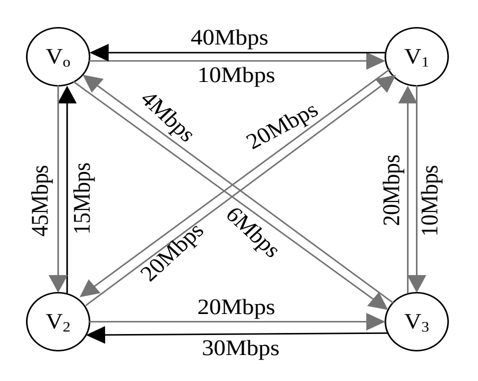
在非对称网络中，节点之间上行和下行的带宽不一致称为非对称链路。网络中的链路大部分都是非对称链路，Lee 等人[24]指出只有 21.49%边可以认为双向链路是对称的。假如把所有链路看成是对称链路，例如图 1(a)中最优再生树的瓶颈带宽是 30 Mbps。但是如果把链路看成非对称链路，如图 1(b)所示，瓶颈带宽仅能达到 15 Mbps。因此把链路带宽看成是非对称链路更符合实际网络情况，在非对称网络中构造最优再生树更能真实地提高瓶颈带宽，图 1(c)中最优再生树的瓶颈带宽可以达到 20 Mbps。非称链路单棵树的再生过程与对称链路下的修复过程类似，该方法存在的问题与对称链路下的修复存在的问题类似。



(a)对称链路网络中的最优再生树



(b)非对称链路网络中的再生树

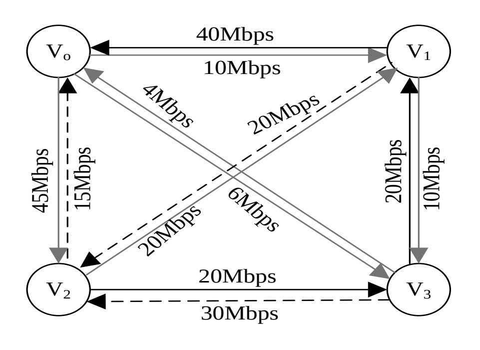


(c)非对称链路网络中的最优再生树

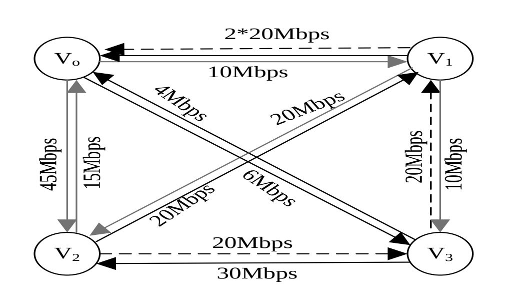
图 1　对称/非对称链路网络中的单棵再生树

2.2.3 非对称链路多棵树并行传输的再生过程

在非对称链路中，构造多棵树并行传输能进一步降低再生时间，因此构造多棵再生树能使用更多链路带宽，降低再生时间，提高再生效率。例如，图 2(a)为两棵树并行再生过程中利用到了 5 条链路，相比于单棵树再生过程，进一步降低了再生时间。另外，如果允许多棵再生树中的边共享网络中的链路，例如，图 2(b)中两棵树同时负责各一半的再生流量，那么瓶颈带宽能提高到 30 Mbps。虽然多棵树能充分利用可用带宽，但是仍然存在类似的问题。



(a)非对称链路网络中的两棵并行再生树



(b)非对称链路网络中的两棵边相交并行再生树

图 2　非对称链路网络中的两棵再生树

2.2.4 针对存储系统以纠删码为编码方式的流水线修复技术

虽然系统使用纠删码产生冗余数据提高了存储效率，但是存在高修复成本的缺点。具体来说，修复一个不可用的编码块需要读取多个可用编码块。与正常读取相比，读取额外的数据块不仅增加了读取时间，也消耗了其他前台服务器的带宽资源。因此，在实践中，纠删码方式主要用于存储不经常需要读取的数据，即冷数据，而经常需要读取的数据，即热数据，以副本方式存储。副本方式仅需要简单地从其他可用节点上读取相应的副本，这样可以保持高效的访问速度。为了减少纠删码的修复时间，许多研究工作或者提出了新的编码方案，或者设计新的修复方法，虽然这些方法有效地减少了修复时间，但修复时间仍然高于一般的正常读取时间。

基于此，Li 等人[25]提出了一种新的流水线修复技术（repair pipelining），这种新的流水线修复技术可以同时应用在同构环境下和异构环境下。他们的做法是把一个失效块的修复过程转化为若干个片的修复过程。具体来说，就是把块平均分成 s 个大小相等的片，例如(14,10)RS 编码系统中，设置一个块的大小为 64 MiB，流水线修复失效块时，把失效块分成 2 048 个片，每个片大小为 32 KiB，各个片的修复过程按流水方式进行。修复时间用时间戳来定义，时间单位用时隙表示。

在同构环境下，利用流水修复技术可以快速解决单个条带内一个数据块的降级读问题，修复时间为 $1+\frac{k-1}{s}$ 个时隙。从这个表达式可以看出，s 越大，修复时间趋近于一个时隙，这意味着降级读时间接近正常读时间。不仅如此，流水修复技术还解决了多条带上的修复问题。修复不同条带内的块时，使用贪婪调度法给每个节点增加一个表示时间戳的符号，用来追踪其最近一次被选为供应节点的时间，目的是均衡各个供应节点的负载。图 3 所示为 RS(6,4)编码系统中修复一个块的过程，把失效块分成 6 个片，表示为 $S_1,S_2,\ldots,S_6$，每 3 个片为一组，用 Group1 和 Group2 表示 2 个组。修复一个数据块的过程中，第一阶段消耗 0.5 个时隙，第二阶段每条路径的最后一个供应节点同时传送片至请求者(R)，同样消耗 0.5 个时隙。

异构环境下，每对节点的链路带宽不一样，对于路径的选择，文献提出了一个带权路径最优选择算法，该算法可以在极短时间内找出带宽最大的一条路径，例如，(14,10)MDS 编码系统，用枚举法搜索所有路径找出最优路径平均需要 27s, 而用文献提出了算法搜索最优路径仅需 0.9 s。

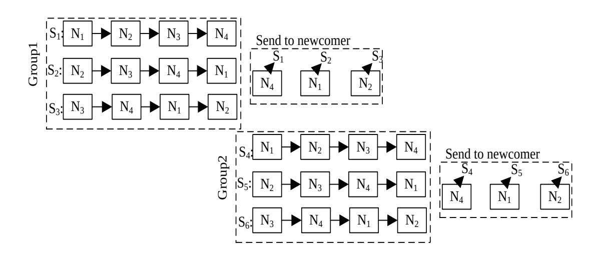


图 3　流水线修复过程,其中 k=4,s=6

在同构环境下，流水线修复的修复时间达到 $O(1)$ 。异构环境下，如果系统采用经典 Reed Solomon 码，其修复时间能达到 $O(k)$ 。最近提出的 PPR 修复方法的修复时间能达到 $O(\log^k)$ 。在许多情况下，相比于传统修复方法和 PPR 修复方法，流水线修复单个块的修复时间能减少到 80%到 90%，提高了基于 HDFS 和 QFS 部署的系统修复性能。

2.2.5 基于传输成本异构的数据修复方法

传输成本是指相邻节点单条链路传输一个元素的成本，不同链路上的传输成本不总是相等。另外，不同类型的网络拓扑结构也会影响总修复成本，进而影响修复方案的选择。基于此，Akhlaghi 等人[26]考虑了比较简单的场景，假设系统中有两类节点 S_1 和 S_2，每类节点都对应着不同的通信代价 C_1 和 C_2。根据再生码要求修复时新生节点要连接 d 个供应节点的要求，假设从通信代价为 C_1 的节点集合中选取 d_1 个节点，从通信代价为 C_2 的节点集合中里选取 $d_2=d-d_1$ 个节点。此时，修复一个新生节点所需的总修复代价为 C_T 可以表示为：$C_T=(C_1d_1+C_2d_2)\beta$ 。在此基础上他们给出了修复代价和修复带宽的权衡关系。该方法的局限性在于，系统里只有两类通信代价的节点，而实际系统中通信代价可能多种多样，另外，也没有考虑网络拓扑结构对修复过程的影响。李钧等人考虑了网络结构对修复过程的影响的影响，提出了再生码的最优树型修复策略。Gerami 等人[27]考虑了四种典型的网络模型：串行网络、星型网络、网格以及全联通网络，提出可用节点合作再生的方案（SNC）以减少修复成本，提出联合法和分离法优化以修复成本。联合法的主要思想是构造修复成本最小的 (n,k,α,d,β) 再生码，且该再生码满足 MDS 特性；分离法是利用 MDS 特性，通过分析信息流图，找出可行域，把优化修复成本问题转换为线性规划问题。使用 SNC 方案修复失效的数据可以充分利用网络的拓扑结构，达到减少修复成本的目的，他们研究的局限性在于，他们使用了和典型再生码一样的假设，即新生节点从每个供应节点下载的数据量相等。

2.3　基于 XOR 的纠删码修复方法

基于异或操作(XOR)纠删码的低网络负载数据修复技术[28]最早由 Xiang Liping 等人提出，用于优化 RDP 码[29]修复所传输的数据量， Khan 等人将其一般化，使其适用于任何基于异或操作纠删码[30]。基于异或操作纠删码的每一个数据块可以看成由多个大小相等的数据片组成，编码块中的编码片经过某些数据片异或运算而产生。下面介绍三个基于异或操作纠删码

的数据修复技术，分别是提高降级读性能的 EG 算法修复策略、基于 RAID-6 编码的 PHR 算法修复策略和基于下载成本异构的数据修复技术，并分析其性能。

2.3.1 提高降级读性能的 EG 算法修复方法

节点失效的类型有两种，永久失效和暂时失效，永久失效是指节点存储的数据丢失了，暂时失效是指节点存储的数据没有丢失只是暂时不能读。对于前一种失效情况，系统进行失效修复（failure recovery），对于后一种，系统进行降级读（degrade read），降级读操作既要读可用数据，也要读不可用的数据，如果需要读不可用数据时，系统要进行相应数据的恢复操作。使用纠删码方式进行编码的过程可以用矩阵来表示，当单个节点失效时，可用节点存储的编码块仍然可以用一个编码子矩阵表示，恢复数据的过程就是计算相应解码子矩阵的过程。考虑到节点带宽的异构性，Zhu 等人[31]提出了枚举贪心算法（enumerated greedy algorithm），简称 EG 算法，EG 算法遍历所有 d 个可用节点可能的组合，在每一种组合下有 l 个 CDREs，计算每个 CDREs 的降级读时间，更新降级读时间，使每个块的降级读时间最小。

假定一个(k,m,w)纠删码系统，系统中有 $n(n=k+m)$个存储节点，k 表示数据节点数量，m 表示校验节点数量，w 表示一个条块中的编码块数量。假设 f 个节点失效，根据 MDS 特性，从 $d(d \le n-f)$ 个可用节点中选择任意 k 个节点可以得到一个解码子矩阵，因此每个数据块有 $\binom{d}{k}$ 个 CDREs。如果要完成 $l(0 \le l \le kw-1)$ 个数据的降级读取请求，那么需要用 l 个 CDREs 再生 1 个数据块，因此解空间域是 $\binom{d}{k}^l$。文献提出的 EG 算法能在合理的时间内找出最优的降级读序列，其时间复杂度为 $O(\binom{d}{d-k}l)$,相比于基本法（basic approach）,EG 算法可以减少降级读取时间 32.70%。

2.3.2 基于 RAID-6 编码的 PHR 算法修复方法

RAID 编码或者叫冗余磁盘阵列技术，目前已经成为一项重要工业标准，基于副本和纠删编码的各种 RAID 技术为海量数据的存储提供了更高可靠性保障。该技术主要利用条带技术（并行 I/O 技术）和冗余技术分别使存储系统中的数据能并行存取和恢复，从而实现磁盘存储系统的高性能、高可靠性和大容量。最初磁盘阵列主要包括 RAID0 到 RAID5，随着大规模存储系统对可靠性提出了更高的要求，容错能力为 2 的 RAID-6 编码被提出了，例如 EVENODD 编码、RS 编码和 RDP 编码等。

RAID6 采用专用的双校验磁盘($P+Q$)，即行校验磁盘和对角校验磁盘。行校验磁盘中的元素由所在行各数据元素 XOR 运算得到，对角校验磁盘中的元素由所在对角线上所有元素 XOR 运算得到。图 4 所示为 RDP 编码系统，p=5。假设 *Disk0* 失效，采用不同的数据恢复策略，读取的数据元素总量也不相等。采用传统修复方法需读取 16 个元素，而采用混合修复策略仅需读取 12 个元素,读取量降低了 25%。

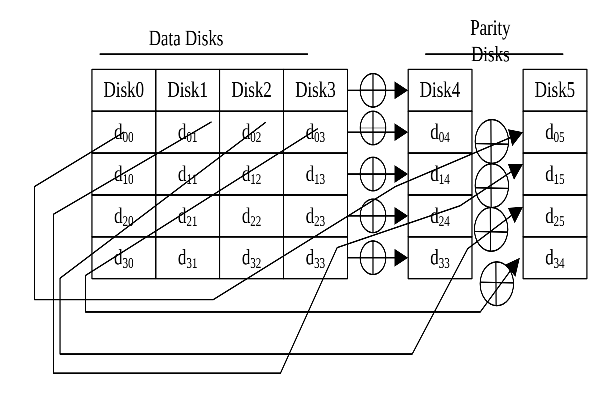


图 4　由 6 个磁盘组成的 RDP 编码系统,其中 p = 5

Niu 等人[32]提出了 RAID-6 编码系统中多条带修复策略。他们把单个条带的修复过程划分为三阶段：读数据阶段、解码数据阶段、写入数据阶段，也就是图 5 所示的单条带修复过程。利用 Holland 等人[33]所提出的多线程技术，同时考虑了节点异构性，提出了并行异构恢复算法（parallel heterogeneous recovery,PHR），该算法能及时返回一个最优修复序列。使用 PHR 算法修复单条带上的失效数据时，把修复过程划分成三个阶段，这三个阶段如下：根据 PHR 算法返回的修复序列，从可用磁盘上读取相应的数据元素或校验元素;解码已读取元素得到丢失的元素;写入已解码元素到其他磁盘上。这三个阶段按顺序进行。由于各个条带的修复过程是相互独立的，因此可以借助多线程和流水线技术同时执行修复操作，也就是说在修复不同条带上的失效数据时，可以并行化 PHR 算法，达到进一步降低再生时间的目的。图 6 描述了多条带修复过程，R_i,D_i,W_i 分别代表条带 i 上的读阶段、解码阶段和写入阶段，对于条带 i 上的读阶段，磁盘的数量等于线程的数量，每个线程 b 独立执行读操作，而在解码阶段，线程的数量取决于 CPU 的速度。

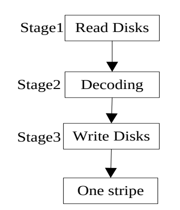


图 5　一个条带上的修复过程

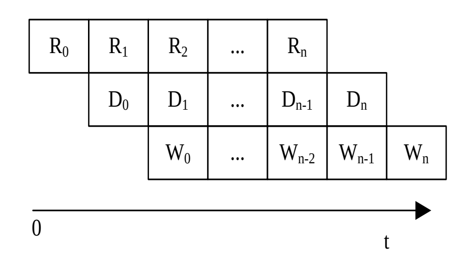


图 6　多条带上的修复过程

相比于枚举法搜索最优修复序列，PHR 算法搜索最优修复序列效率更高。例如 p=23 时，枚举法搜索最优算法需花 17s，而 PHR 算法仅需 3s。修复多个条带时，并行化执行

PHR 算法进一步降低了修复时间。

2.3.3 基于下载成本异构的数据修复技术

传输成本是指一条链路上传输一个元素的成本，节点之间带宽资源不同，传输成本也不同。基于此，Zhu 等人[34]把传输成本和带宽资源联系起来，定义修复总成本为 $C=\sum_{i=0,i\neq k}^{p} w_i y_i$，其中假设节点 k 失效，w_i 表示从节点 i 读一个元素的成本，y_i 表示从节点 i 读 y_i 个元素。他们提出基于传输成本的异构恢复（cost-based heterogeneous recovery，CHR）。CHR 算法枚举所有可能的最小读取量恢复序列，计算这些序列的总修复成本，返回最小总修复成本对应的修复序列。CHR 算法把那些相反或反向最少读取量恢复序列归为一种最优恢复序列，从而缩小遍历空间，减少遍历所有读取量最少的恢复序列的计算开销。

图 7 所示为异构环境下 RDP 编码系统，其中 p=7,各个节点的带宽如图所示。假设读取一个元素的成本为 1，节点 0 失效。用传统修复方法修复节点 0，Proxy 从前 6 个节点中分别读取 6 个元素,其总下载成本是 $0.9921\alpha(in\sec)$；用混合修复方法修复节点 0 得到其中的一个最优修复序列{1110000}，Proxy 从节点 3、4、7 分别读取 4 个元素，从节点 2、5 分别读取 4 个元素，从节点 1、6 分别读取 5 个元素，其下总载成本是 $0.7353\alpha(in\sec)$；用 CHR 算法搜索得到的其中一个最优恢复序列为{1010100}，各个节点传输的元素如图 8 所示，Proxy 从节点 1、6、7 分别读取 3 个元素，从节点 2、5 分别读取 5 个元素，从节点 3、4 分别读取 4 个元素,其总成本是 $0.5449\alpha(in\sec)$。该方法比传统修复方法的下载成本少 40.91%，比混合修复方法的下载成本少 25.89%。

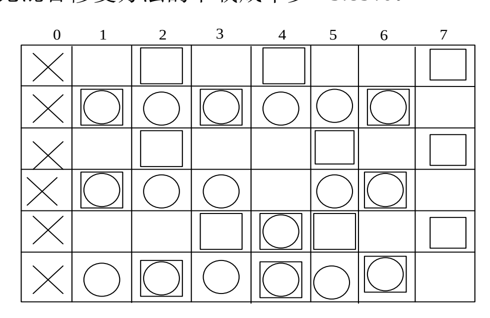


图 7 异构环境下 RDP 编码系统，其中 $p=7$

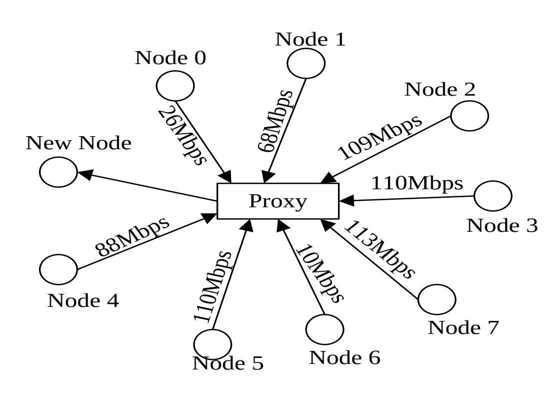


图 8　CHR 算法修复方案

在异构环境下，CHR 算法能高效地修复单个失效节点，遍历效率、鲁棒性效率、恢复效率高。

3　面向计算能力异构的修复方法

在分布式存储系统中,每个存储节点由于自身的各种因素会造成其对数据的处理速度不一样，我们称这种差异为节点的计算能力异构。在进行数据修复的过程中，非叶子节点需要读取本地数据，并结合接收到的数据进行编码，将编码后的结果传输给上一个节点。其中，读取本地数据、编码运算等操作的处理速度受制于节点自身的运算速度。因此计算能力异构就体现在节点编码的时间上，节点计算能力越强，处理数据的速度越快。Henry[35]的调研表明磁盘 I/O 已成为存储节点对本地数据的读写瓶颈,所以节点的计算能力异构对数据修复的影响是不可忽略的，然而却很少有人研究这方面的问题，下面仅介绍一篇有关节点计算能力异构的文献，即分布式存储再生码数据修复的节点选择方案，并讨论其在修复带宽开销、修复时间开销和参与修复的节点数量 3 个方面的表现。

3.1　分布式存储再生码数据修复的节点选择方案

李钧等人考虑了带宽资源异构对修复过程的影响，假设节点在修复过程中以流水线方式并行传输数据，忽略数据在节点处的处理时间，提出树形修复策略以提高瓶颈带宽，从而减少修复时间。然而在实际的分布式存储系统中，节点的处理时间对修复过程的影响很大，仅仅提高瓶颈带宽，忽略节点处理时间不一定能减少修复时间。齐凤林等人[36]同时考虑了节点计算能力异构和带宽资源异构对数据修复过程的影响，建立星型和树型修复拓扑结构。对于供应节点的选择问题，他们分别提出了 S-SPA-C 算法和 T-SPA-C 算法解决该问题。

对于星型修复结构，供应节点直接将数据传输给新生节点，新生节点对接收到的数据进行编码并保存。因此整个修复时间受制于各供应节点中时延最长的节点，修复时间可以表示为 $t=\max\{T_{ci}+T_{Bi0}\}$，其中 T_{ci} 表示节点 i 的处理时延, T_{Bio} 表示从节点 i 传输到新生节点的时间（传输时延）。构造星形修复结构的做法是利用节点的计算时延和传输时延之和大小确定供应节点，基本做法是计算除失效节点外 $n-1$ 个节点的计算能力大小 c，并结合传输量,计算这些节点到新生节点的传输时延，求得 $n-1$ 个节点对数据的处理时间与相应传输到新生节点的传输时延之和 $T=T_{ci}+T_{Bi0}$。把 T 按照从小到大的顺序排序编号，取前编号为 1,2,...,d 的节点作为供应节点。以图 9 为例，图中有 4 个供应节点 P_1,P_2,P_3,P_4 和一个新生节点 P_0,图中圆圈内的数据表示节点的处理时间，边上的数据表示节点间的传输时延。假设新生节点连接 3 个节点修复失效数据，通过 S-SPA-C 算法确定了 P_1,P_3,P_4 为供应节点，修复过程如图 10 所示，此轮修复时间是 7.4t。

对于树型修复结构，叶子节点将数据向其父节点传输，非叶子节点从其下级子节点接收数据，并结合其自身的编码块编码后将结果向上传输给其父节点，逐级上传至根节点（新生节

点），根节点接收到所有数据后编码生成新的编码块并保存。构造树形修复结构的做法是利用最小生成树原理确定供应节点，先逐个计算节点从不同链路把数据传输到新生节点所需时间，然后对所需时间从小到大排序，确定各个节点传输数据给新生节点的路径，最终确定修复结构。仍以图仍以图 9 为例，假设取 3 个节点作为供应节点，使用 T-SPA-C 算法得出 P_1,P_3,P_4 为供应节点，其中 P_3 选择路径 $P3 \rightarrow P4 \rightarrow P0$ 将数据传输给 P_0,修复过程如图 11 所示，此时，修复时间由原来的总时间 5.9*t* 减少到 5.4*t*。

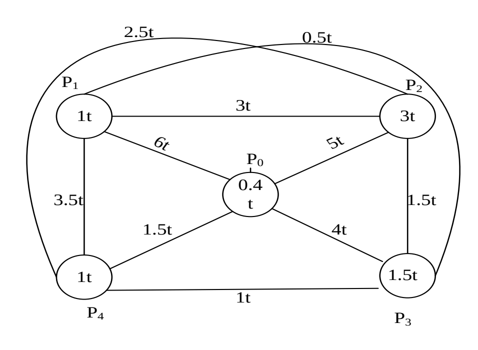


图 9 节点对数据的处理时间及节点间的传输时延

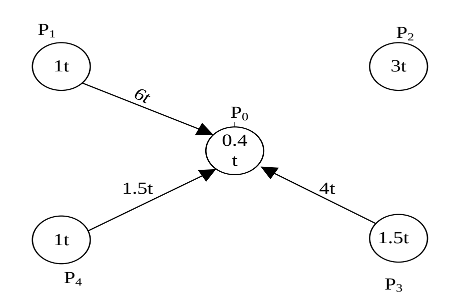


图 10 星型拓扑修复结构

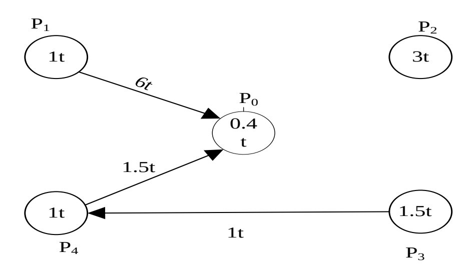


图 11 树型拓扑修复模型

这种选择供应节点的方案在一定程度上可以加快失效数据的修复速度，降低整个数据修复过程的时间，提升整个存储网络的性能。

4 面向存储异构的修复方法

存储容量异构是指每个存储节点上存储的数据量并不总是相等。目前已有关于节点存储容量的研究主要集中在考察节点存储容量的变化对用户成功解码出原文件概率的影响，如文献[37-48]，Leong 等人[44]首先研究了如何将文件分布到各个存储节点上使得用户成功获取文件的概率最大。Li 等人[39]针对分布式存储系统中每个节点以不同概率被成功访问的场景，设计了一种数据分布方法，以提高用户成功获取文件的概率。他们提出的是一种分级均匀分布的数据分布方法，和完全不均匀的数据分布方法[37]相比，这种分级均匀分布的方法能获得更好的性能。李佳等人[49]考虑到了节点的存储性能异构性，提出了基于纠删码的云文件系统数据放置方法，根据节点的实时负载情况进行数据放置，实现了系统的负载均衡并提高了数据写入和修复速度，但是没有进一步研究存储异构对数据修复性能的影响，目前，很少有文献讨论了节点存储量变化对纠删码数据修复的影响，下面介绍仅一种存储容量异构的数据修复技术，并分析该方法在修复带宽开销上的表现。另外，介绍了几篇篇有关存储容量变化的文献。

文献[50]描述了这样一个存储系统，系统中存在一个超级节点，该超级节点的存储量比其他节点大，可靠性和可用性也比其他节点高。该文献针对$(k+2,k)$MDS 码和$(k+2,k)$非 MDS 码提出了三种分配存储方案，超级节点存储 2α 个块,其他节点都存储 α 个块，每种存储方案下的存储内容不一样。在每种存储方案下，文献考虑了节点所有可能失效的情况，对每一种失效情况描述了修复过程，最后分析了三种存储方案方式下数据的可靠性。从修复带宽和数据可靠性角度分析，修复一个失效节点，相比于传统分配方式（所有节点的存储量都相同），文献提出的三种分配方式下的数据修复带宽能达到最小带宽 $\frac{M(k+1)}{2k}$,修复两个节点仅需 $\frac{M}{2k}$,数据可靠性能提高了 10%。

文献[13]通过分析信息流图，获取到了存储量和修复带宽的权衡关系。他们证明，如果该信息流图中的最小割（min-cut）大于原文件的大小，则存在线性编码使得每个 DC 节点都可以恢复出原文件，如果随机线性编码在有限域充分大时可以使得 DC 节点以接近 1 的概率恢复出原文件。

图 12 给出了参数为 *n*=10，*k*=5，*d*=9 的存储和带宽开销的权衡关系。该曲线表明单个节点储存量 α 越大，修复单个失效节点的带宽开销 γ 就越少。存储和带宽开销满足该曲线关系的编码称为再生码（regenerating codes）。曲线上的两个极值点分别对应着两类特殊的编码：最小存储再生码（minimum storage regenerating codes，MSR 码）和最小带宽再生码（minimum bandwidth regenerating codes，MBR 码），它们分别对应不同的(α,γ)值：

$$(\alpha_{MSR},\gamma_{MSR}) = (\frac{M}{k},\frac{M}{k}\cdot\frac{d}{d-k+1}) \tag{3}$$

$$(\alpha_{MBR},\gamma_{MBR}) = (\frac{M}{k}\cdot\frac{2d}{2d-k+1},\frac{M}{k}\cdot\frac{2d}{2d-k+1}) \tag{4}$$

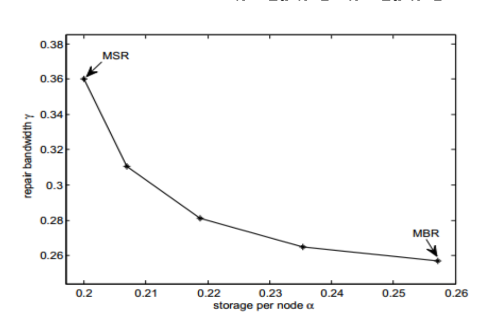


图 12 参数为 n=10,k=5,d=9 的存储和带宽开销的权衡曲线

Dimakis 等人得出了单节点修复模型下节点存储与修复带宽的理论界,对于多节点修复模型，Shum 等人[51]提出了新生节点之间相互合作的模型，并给出此模型下存储与带宽的理论界，Zhang 等人[52]提出新生节点之间不再相互传输数据的模型，比合作修复减少了设计和运算的复杂性，更符合系统的实际需要，王丽莎等人[53]针对这种新的修复模型，利用割原理找出其最小割，并用线性规划的方法给出存储和带宽的理论界，过程更为简单，在此基础上，给出一些特殊参数下的编码构造的方法。李松涛等人[54]针对数据存储成本和带宽成本，提出了一种称为缓存大小自适应确定(CAROM)的新方案，该方案结合传统的基于缓存策略的方法和纠错码方法来提高系统的修复效率。为了实现缓存大小及其效益间平衡，提出一种基于总体成本凸函数特性的弹性方法来实现缓存大小的弹性选择。CAROM 方案的存储成本和带宽成本分别比复制策略和纠错码策略下降 60%和 43%。兼具带宽成本低、存储成本低等特性。

文献[55]考虑了比较简单的场景，假设系统中有两类节点 S_1 和 S_2,每类节点都对应着不同的存储成本 C_1 和 C_2，两类节点的存储量分别为 α_1 和 α_2。假定存储成本为 C_1 的节点数是 n_1,存储成本为 C_2 的节点数是 n_2，此时，系统总存储成本 C_s 可以表示为：$C_s = C_1 n_1 \alpha_1 + C_2 n_2 \alpha_2$。修复失效数据时，新生节点连接任意 d 个供应节点，从每个节点下载 β 数据量，修复带宽 γ 可以表示为：$\gamma = d\beta$。在此基础上他们给出了存储成本和修复带宽的权衡关系。该方法的局限性在于，系统中只有两类存储成本的节点，而实际系统中存储成本可能多种多样。图 13 是一个基于 MDS 码的信息流图。图中显示的是一个参数为 n=4，k=2，d=3 的分布式存储系统，前两个节点的存储成本为 1，存储量都是 α_1,后两个节点的存储成本为 2，存储量为 α_2。假设 $V_2^{in}\rightarrow V_2^{out}$ 是已经失效的存储节点，$V^{in}\rightarrow V^{out}$ 是新加入的存储节点（newcomer）。新加入的存储节点为了完成数据修复，需要从系统中剩余的其他 d 个存储节点分别读取 β 数据量，也就是图中标记的再生流（regeneration traffic）。

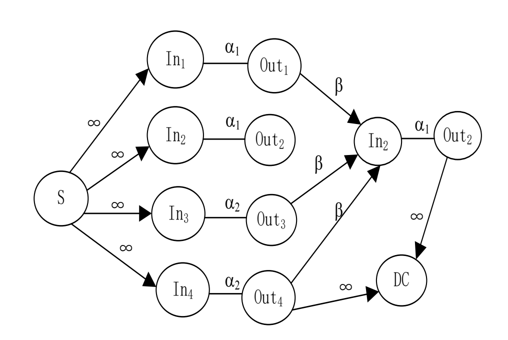
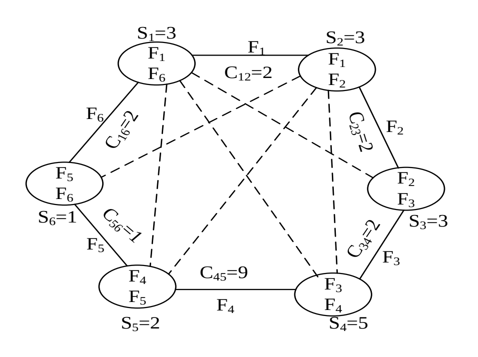


图 13　(4,2)MDS 码的信息流图 n1=n2=2,d=3,k=2,C1=1,C2=2

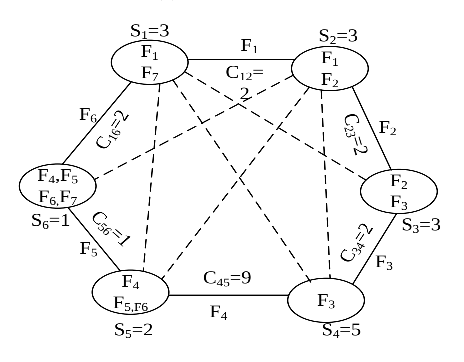
文献[56]将优化存储成本问题推广到更一般的场景，他们假设存储节点 v 的存储量为 α_v (v=1,2,...,n)，存储成本为 s_v(v=1,2,..,n)，则存储一个块的平均成本（系统成本）可以表示为 $\mathbf{c_s}(\tau(\mathbf{x}),\mathbf{b})\overset{\Delta}{=}\frac{1}{B}\sum_{v=1}^{n}s_v\alpha_v\leqslant \mathbf{C_s}$。

他们也考虑了通信代价对数据修复的影响，提出了构造 IFR 码的方法，通过优化 IFR 码以优化数据分配。他们提出的是一种非均匀分布的数据方法，和完全均匀分布的方法[57]相比，这种非均匀分布的方法能获得更好的性能。该方法的局限在于，构造出来的 IFR 编码系统不满足 MDS 特性。纠删码中的 MDS 性质是一个比较好的性质，这个性质可以保证用户最大概率地恢复原文件。如果系统采用 IFR 码产生冗余数据，DC 节点只能通过指定的节点集合恢复原文件，不能满足任意性。

图 14 给出了构造 FR 和 IFR 码的例子，对应的参数都是 n=4,k=2,d=2。如果使用 MDS-FR 码，假设原文件 M=4，经过(6,4)MDS 编码后产生 6 个编码块 $F_1,F_2,...,F_6$，并分配到 6 个节点上。在这个 6 个节点的环中，相邻节点边上的编码块表示这两个节点存储的公共块，边上的数字表示相邻节点相互通信的通信成本。如图 14(a)所示，每个节点的存储量都是 2，节点 1 存储 F1 和 F6，节点 2 存储 F1 和 F2，节点 3 存储 F2 和 F3，节点 4 存储 F3 和 F4，节点 5 存储 F4 和 F5，节点 6 存储 F5 和 F6，总存储成本是 34。任何一个节点失效都可以通过连接其相邻的两个节点修复出相应的数据，总修复成本是 36。如果使用 MDS-IFR 码，假设原文件 M=4，经过(7,4)MDS 编码后产生 7 个编码块 $F_1,F_2,...,F_7$，每个节点储存量如图 14(b)所示，总存储成本是 33，总修复成本是 22。由此可见，后者的分配方式能获得更好的性能，但局限性在于，需要用以比 3 副本更高的存储成本为代价提供非常低的修复成本。



(a)FR 码构造实例



(b)IFR 码构造实例

图 14　FR 码和 IFR 码构造的实例

5　讨论

现有的 (n,k,d,α,β) 纠删码数据修复方案大多是在固定 α 和 β 两个参数的基础上优化修复带宽、修复时间、磁盘访问、参与修复的节点数量和修复代价。这一类的研究工作涵盖了纠删码的功能性修复和精确性修复，但没有考虑到系统中带宽资源、计算资源以及存储资源等对数据修复的影响。然而，实际情况是，在大规模的数据中心里，设备替换、硬件故障等原因不仅仅会导致数据丢失，还会导致数据中心的各个存储节点在硬件上的不同，比如存储节点间会出现可用带宽、计算能力和存储容量的差异。因此研究优化分布式存储编码在异构环境下的冗余数据修复性能具有

重要理论意义和实际意义。

目前，存储系统采用的纠删码大部分是线性随机编码,修复失效数据是通过对可用数据进行线性组合完成，可用数据的选择和相应组合系数由具体的纠删码类型决定。纠删码数据修复技术面临的挑战主要表现在计算、读写和传输 3 个方面。其中，通过 SIMD 技术可以现实对多个数据单元同时执行相同的操作，加快了基于有限域运算的编码计算速度，同时调整计算顺序，避免重复计算，可以有效地降低计算量，来应对计算方面的挑战；通过合理选择修复所使用的条带，让修复所需要的数据出现较多重叠，使得读取的数据块可以用于多个数据块的修复，另外，在不减少数据读取总量的情况下，通过引入更多磁盘到数据修复过程中来，降低单个磁盘上的读取量，以应对读写方面的挑战；通过合理的编码设计，再生码以及其他各种衍生编码可以在带宽开销和供应节点开销等方面提高数据修复性能，某些编码[58-74]支持精确修复，使系统码（支持编码块中包含原数据的编码）有可能应用在分布式系统中，并为数据访问性能的提高提供了技术基础，以应对编码方面的挑战。

为了全面对比现有的纠删码修复性能，表 1 以空间利用率、单块修复代价、总修复代价作为修复性能评价标准，对 6 种典型纠删码的数据修复性能进行了对比。作为对比，表 1 中加入了常见的三副本技术。

表 1　几种典型纠删码与多副本技术的数据修复性能对比

纠删码	空间利用率 / %	单块修复代价	总体修复代价	类别
RS(14,10)	71.40	10.00	14.0	传统 MDS 码
LRCs(10,2,4)	62.50	3.75	6.0	分组码
SHEC(10,6,5)	62.50	5.00	8.00	
(9,5,8)-MSR	55.60	2.00	3.60	再生码
(14,10,13)-MBR	46.70	1.00	1.24	
(14,10)-Hitchhiker-XOR	71.41	7.64	10.70	
三副本	33.30	1.00	3.00	多副本

从表 1 可以看出，没有哪一种编码方案可以很好地满足这 3 个指标，传统 MDS 码 RS(14,10)存储空间利用率最高，但是其单块修复代价和总体修复代价也最大，甚至高于其它种类纠删码数倍。相比于传统 MDS 码，分组码 LRCs(10,2,4)、SHEC(10,6,5)能够以较少的额外存储空间开销为代价，显著降低单块修复和总体修复的成本。再生码(14,10,13)MBR 在单块修复代价和总体袖套代价上都取得良好的表现，但是再生码的存储空间利用率明显低于其它类别纠删码，其存储空间利用率最高也只能达到 50%左右。所以，那些网络带宽成本和存储成本高的系统，比较适合使用此类再生码。

6　结束语

本文分析了影响纠删码数据修复的因素，从带宽资源、计算资源、存储容量资源三方面对优化纠删码修复性能的方法进行了探讨。现有的修复方法大多没有考虑到存储节点存在带宽资源异构、计算资源异构、存储容量资源异构情况。

目前对分布式存储系统中存储编码的研究大多考虑的是同构环境（无差别对待每一个存储节点），即分布式系统中存储节点的带宽资源、计算资源、存储资源都一致，但实际情况是，因为地理差异和磁盘性能的不同，会导致各个节点硬件上的不同，即便有少量针对异构环境纠删码数据修复的研究工作也是集中在带宽资源异构对纠删码数据修复的影响，少有考虑节点计算能力和存储能力异构对纠删码数据修复的影响。另外，目前纠删码数据修复技术在带宽开销、时间开销等方面都不同程度地存在着较大的缺陷，难以同时使得这些目标都达到理想的状态。于是异构分布式系统在实际部署中变得非常有意义，有关异构分布式系统的数据修复技术仍然停留在理论上的研究，实际应用仍然是一片空白。我们未来会将最优理论中的一些修复方法和图论中一些性质应用到实际的分布式系统中，并研究一些特殊场景下编码后修复性能增益的问题。至于如何设计出各方面俱优的纠删码数据修复技术仍是未来研究中任重道远的问题。

参考文献：

[1] Lakshmi N. Bairavasundaram, Garth R, *et al.* An analysis of latent sector errors in disk drives [J]. ACM SIGMETRICS Performance Evaluation Review, 2007, 35 (1): 289-300.

[2] Honnutagi P S. The Hadoop distributed file system [J]. International Journal of Computer Science & Information Technolo, 2014, 5 (5): 6238-6243.

[3] Weil S A, Brandt S A, Miller E L, *et al.* Ceph: a scalable, high-performance distributed file system [C]// Proc of the 7th Conference on USENIX Symposium on Operating Systems Design and Implementation. Berkeley: USENIX Association, 2006: 307-320.

[4] Reed I S, Solomon G. Polynomial codes over certain finite fields [J]. Journal of the Society for Industrial & Applied Mathematics, 1960, 8 (2): 300-304.

[5] Rashmi K V, Shah N B, Gu Dikang, *et al.* A solution to the network

challenges of data recovery in erasure-coded distributed storage systems: a study on the Facebook warehouse cluster [C]// Proc of the 5th Workshop on Hot Topics in Storage and File Systems. Berkeley: USENIX Association, 2013: 1-5.

[6] Dimakis A G, Ramchandran K, Wu Yunnan, *et al.* A survey on network codes for distributed storage [J]. Proceedings of the IEEE, 2011, 99 (3): 476-489.

[7] Li Jun, Li Baochun. Erasure coding for cloud storage systems: a survey [J]. Tsinghua Science and Technology, 2013, 18 (3): 259-272.

[8] 罗象宏，舒继武. 存储系统中的纠删码研究综述 [J]. 计算机研究与发展, 2012, 49 (1): 1-11. (Luo Xianghong, Shu Jiwu. Summary of research for erasure code in storage system [J]. Journal of Computer Research and Development, 2012, 49 (1): 1-11.)

[9] 王意洁，许方亮，裴晓强. 分布式存储中的纠删码容错技术研究 [J]. 计算机学报, 2017, 40 (1): 236-255. (Wang Yijie, Xu Fangliang, Pei Xiaoqiang. Research on erasure code-based fault-tolerant technology for distributed storage [J]. Chinese Journal of Computers, 2017, 40 (1): 236-255.)

[10] 杨松霖，张广艳. 纠删码存储系统中数据修复方法综述 [J]. 计算机科学与探索, 2017, 11 (10): 1531-1544. (Yang Songlin, Zhang Guangyan. Review of data recovery in storage systems based on erasure codes [J]. Journal of Frontiers of Computer Science and Technology, 2017, 11 (10): 1531-1544.)

[11] Ernvall T, El Rouayheb S, Hollanti C, *et al.* Capacity and security of heterogeneous distributed storage systems [J]. IEEE Journal on Selected Areas in Communications, 2013, 31 (12): 2701-2709.

[12] Pei Xiaoqiang, Wang Yijie, Ma Xingkong, *et al.* Cooperative repair based on tree structure for multiple failures in distributed storage systems with regenerating codes [C]// Proc of the 12th ACM International Conference on Computing Frontiers. New York: ACM Press, 2015: 784-792.

[13] Dimakis A G, Godfrey P B, Wu Yunnan, *et al.* Network coding for distributed storage systems [J]. IEEE Trans on Information Theory, 2010, 56 (9): 4539-4551.

[14] Wu Yunnan, Dimakis A, Ramchandran K. Deterministic regenerating codes for distributed storage [C]// Proc of Allerton Conference on Control, Computing, and Communication. 2007.

[15] Shah N B, Rashmi K V, Kumar P V. A flexible class of regenerating codes for distributed storage [C]// Proc of IEEE International Symposium on Information Theory. 2010: 1943-1947.

[16] Gong Qingyuan, Wang Jiaqi, Wei Dongsheng, *et al.* Optimal node selection for data regeneration in heterogeneous distributed storage systems [C]// Proc of International Conference on Parallel Processing. 2015: 390-399.

[17] Akhlaghi S, Kiani A, Ghanavati M R. A fundamental trade-off between the download cost and repair bandwidth in distributed storage systems [C]// Proc of IEEE International Symposium on Network Coding. 2010: 1-6.

[18] 贾亚男，岳殿武. 博弈论框架下认知小蜂窝网络的动态资源配算法 [J]. 电子学报, 2015, 43 (10): 1911-1917. (Jia Yanan, Yue Dianwu. Dynamic resource allocation algorithm based on game theory in cognitive small cell networks [J]. Acta Electronica Sinica, 2015, 43 (10): 1911-1917.)

[19] 洪浩，张焱，肖立民，等. 认知双向中继网络的功率分配优化算法研究 [J]. 电波科学学报, 2014, 29 (2): 201-206+226. (Hong Hao, Zhang Yan, Xiao Limin, *et al.* Optimal power allocation for cognitive two-way relaying networks with underlay spectrum sharing [J]. Chinese Journal of Radio Science, 2014, 29 (2): 201-206+226.)

[20] 郑力明，李晓冬. 面向纠删码的低成本多节点失效修复方法 [J]. 计算机工程, 2017, 43 (7): 110-118, 123. (Zheng Liming, Li Xiaodong. Low-cost multi-node failure repair method for erasure codes [J]. Computer Engineering, 2017, 43 (7): 110-118, 123.)

[21] Li Jun, Yang Shuang, Wang Xin, *et al.* Tree-structured data regeneration with network coding in distributed storage systems [C]// Proc of International Workshop on Quality of Service. 2009: 2892-2900.

[22] Li Jun, Yang Shuang, Wang Xin, *et al.* Tree-structured data regeneration in distributed storage systems with regenerating codes [C]// Proc of INFOCOM. 2010: 1-9.

[23] Wang Yan, Wei Dongsheng, Yin Xunrui, *et al.* Heterogeneity-aware data regeneration in distributed storage systems [C]// Proc ofINFOCOM. 2014: 1878-1886.

[24] Lee S J, Sharma P, Banerjee S, *et al.* Measuring bandwidth between planetlab nodes [C]// Proc of International Conference on Passive and Active Network Measurement. Springer-Verlag, 2005: 292-305.

[25] Li Runhui, Li Xiaolu, Lee P P C, *et al.* Repair pipelining for erasure-coded storage [C]// Proc of Usenix Technical Conference. Berkeley: USENIX Association, 2017.

[26] Akhlaghi S, Kiani A, Ghanavati M R. Cost-bandwidth tradeoff in distributed storage systems [J]. Computer Communications, 2010, 33 (17): 2105-2115.

[27] Gerami M, Xiao Ming, Skoglund M. Optimal-cost repair in multihop distributed storage systems [C]// Proc of IEEE International Symposium on Information Theory Proceedings. 2012: 1437-1441.

[28] Xiang Liping, Xu Yinlong, Lui J C S, *et al.* Optimal recovery of single disk failure in RDP code storage systems [J]. ACM SIGMETRICS Performance Evaluation Review, 2010, 38 (1): 119-130.

[29] Corbett P, English B, Goel A, *et al.* Row-diagonal parity for double disk failure correction [C]// Proc of Usenix Conference on File and Storage Technologies. Berkeley: USENIX Association, 2004: 1-1.

[30] Khan O, Burns R, Plank J, *et al.* Rethinking erasure codes for cloud file systems: minimizing I//O for recovery and degraded reads [C]// Proc of Usenix Conference on File and Storage Technologies. Berkeley: USENIX Association, 2012: 20.

[31] Zhu Yunfeng, Lin Jian, Lee P P C, *et al.* Boosting degraded reads in heterogeneous erasure-coded storage systems [J]. IEEE Trans on Computers, 2015, 64 (8): 2145-2157.

[32] Niu Fang, Xu Yinlong, Zhu Yunfeng, *et al.* PHR: a pipelined heterogeneous recovery for raid6-coded storage systems [C]// Proc of International Conference on Parallel and Distributed Computing, Applications and Technologies. 2014: 325-331.

[33] Holland M, Gibson G A, Siewiorek D P. Architectures and algorithms for on-line failure recovery in redundant disk arrays [J]. Distributed & Parallel Databases, 1994, 2 (3): 295-335.

[34] Zhu Yunfeng, Lee P P C, Xiang Liping, *et al.* A cost-based heterogeneous recovery scheme for distributed storage systems with RAID-6 codes [C]// Proc of IEEE//IFIP International Conference on Dependable Systems and Networks. 2012: 1-12.

[35] Henry. I/O bottlenecks: biggest threat to data storage. [2009-12-31]. http://www. enterprisestorageforum. com/technology/features/article. php/3856121/IO-Bottlenecks-Biggest-Threat-to-Data-Storage. html.

[36] 齐凤林，宫庆媛，周扬帆，等. 分布式存储再生码数据修复的节点选择方案 [J]. 计算机研究与发展，2015, 52 (Suppl): 68-74. (Qi Fenglin, Gong Qingyuan, Zhou Yangfan, *et al.* Heterogeneity-aware node selection of data repair in distributed storage systems [J]. Journal of Computer Research and Development, 2015, 52 (Suppl.): 68-74.)

[37] Li Zhao, Ho T, Leong D, *et al.* Distributed storage allocation for heterogeneous systems [C]// Communication, Control, and Computing. 2013: 320-326.

[38] Huang Zhen, Yuan Yuan, Peng Yuxing. Storage allocation for redundancy scheme in reliability-aware cloud systems [C]// Proc of IEEE International Conference on Communication Software and Networks. 2011: 275-279.

[39] Ntranos V, Caire G, Dimakis A G. Allocations for heterogeneous distributed storage [C]// Proc of IEEE International Symposium on Information Theory Proceedings. 2012: 2761-2765.

[40] Kao Y H, Dimakis A G, Leong D, *et al.* Distributed storage allocations and a hypergraph conjecture of Erdős [C]// Proc of IEEE Internation Symposium on Information Theory Proceedings. 2013: 902-906.

[41] Xu Guangping, Lin Sheng, Wang Gang, *et al.* HERO: heterogeneity-aware erasure coded redundancy optimal allocation for reliable storage in distributed networks [C]// Proc of IEEE International Performance Computing and Communications Conference. 2012: 246-255.

[42] Leong D, Dimakis A G, Ho T. Distributed storage allocation problems [C]// Proc of Workshop on Network Coding, Theory, and Applications. 2009: 86-91.

[43] Derek Leong, Alexandros G. Dimakis, Tracey Ho. Distributed Storage Allocation for High Reliability [C]// IEEE, International Conference on Communications. IEEE, 2010: 1-6.

[44] Leong D, Dimakis A G, Ho T. Distributed storage allocations [J]. IEEE Trans on Information Theory, 2012, 58 (7): 4733-4752.

[45] Leong D, Dimakis A G, Ho T. Symmetric allocations for distributed storage [C]// Proc of Global Telecommunications Conference. 2010: 1-6.

[46] Sardari M, Restrepo R, Fekri F, *et al.* Memory allocation in distributed storage networks [C]// Proc of IEEE International Symposium on Information Theory. 2010: 1958-1962.

[47] Hong Tao, Wu Yating, Cao Bingyao, *et al.* A dynamic data allocation method with improved load-balancing for cloud storage system [C]// Proc of IET International Conference on Smart and Sustainable City. 2013: 183-188.

[48] 李君，侯孟书. 基于萤火虫优化的副本放置方法 [J/OL]. 计算机应用研究，2019, 36 (3) . [2018-02-02]. http://www. arocmag. com/article/02-2019-03-045. html. (Li Jun, Hou Mengshu. Replica placement strategy based on glowworm swarm optimization [J/OL]. Application Research of Computers, 2019, 36 (3) . [2018-02-02]. http://www. arocmag. com/article/02-2019-03-045. html.)

[49] 李佳，陈海涛，芦伟. 基于纠删码的云文件系统数据放置方法 [J]. 北京信息科技大学学报：自然科学版，2014, 29 (6): 1-6. (Li Jia, Chen Haitao, Lu Wei. A Novel way of data placement of cloud file system based on erasure code [J]. Journal of Beijing Information Science and Technology University: Natural Science, 2014, 29 (6): 1-6.)

[50] Van Vo T, Chau Yuen, Li Jing. Non-homogeneous distributed storage systems [C]// Communication, Control, and Computing. 2012: 1133-1140.

[51] Shum K W, Hu Yuchong. Cooperative regenerating codes [J]. IEEE Trans on Information Theory, 2013, 59 (11): 7229-7258.

[52] Zhang Huayu, Li Hui, Hou Hanxu, *et al.* Concurrent regenerating codes and scalable application in network storage [J]. arXiv preprint arXiv: 1604. 06567, 2016.

[53] 王丽莎，唐小虎. 新多节点修复模型下的再生码 [J]. 计算机应用研究，2018, 35 (2): 527-531. (Wang Lisha, Tang Xiaohu. regenerating codes for new multi-node repair model [J]. Application Research of Computers, 2018, 35 (2): 527-531.)

[54] 李松涛，金欣. 基于混合策略的低成本云存储方案 [J]. 计算机应用，2014, 34 (10): 2800-2805+2811. (Li Songtao, Jin Xin. Low-cost cloud storage scheme based on hybrid strategy [J]. Journal of Computer Applications, 2014, 34 (10): 2800-2805+2811.)

[55] Yu Quan, Shum K W, Sung C W. Minimization of storage cost in distributed storage systems with repair consideration [C]// Proc of Global Telecommunications Conference. 2012: 1-5.

[56] Yu Quan, Sung C W, Chan T H. Irregular fractional repetition code optimization for heterogeneous cloud storage [J]. IEEE Journal on Selected Areas in Communications, 2014, 32 (5): 1048-1060.

[57] El Rouayheb S, Ramchandran K. Fractional repetition codes for repair in distributed storage systems [C]// Communication, Control, and Computing. 2010: 1510-1517.

[58] Papailiopoulos D S, Luo Jianqiang, Dimakis A G. , *et al.* Simple regenerating codes: Network coding for cloud storage [C]// Proc ofINFOCOM. 2015: 2801-2805.

[59] 徐志强，袁德砦，陈亮. 基于稀疏随机矩阵的再生码构造方法 [J]. 计算机应用，2017, 37 (7): 1948-1952+1959. (Xu Zhiqiang, Yuan Dezhai,

Chen Liang. Regenerating codes construction method based on sparse random matrix [J]. Journal of Computer Applications, 2017, 37 (7): 1948-1952+1959.)

[60] 宋海龙，王伟平，肖亚龙. 基于柯西矩阵的最小带宽再生码研究 [J]. 湖南大学学报：自然科学版, 2017, 44 (8): 152-160. (Song Hailong, Wang Weiping, Xiao Yalong. Study of minimum bandwidth regeneration codes based on cauchy matrix [J]. Journal of Hunan University: Natural Science, 2017, 44 (08): 152-160.)

[61] 曹凯，文捷. 基于 (k+2, k) MSR 的多容错低修复带宽编码 [J]. 计算机工程, 2018, 44 (2): 84-87, 91. (Cao Kai, Wen Jie. Multiple Fault tolerant and low repairing bandwidth coding based on (k+2, 2) MSR [J]. Computer Engineering, 2018, 44 (2): 84-87, 91.)

[62] Qu Shan, Zhang Jinbei, Wang Xinbing. Asymmetric regenerating codes for heterogeneous distributed storage systems [C]// Proc of IEEE International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks. 2018: 1-8.

[63] 万武南，杨威，陈运. 一种新的 3 容错扩展 RAID 码 [J]. 北京邮电大学学报，2014, 37 (5): 75-79. (Wan Wunan, Yang Wei, Chen Yun. A toleration based extended raid code triple failures [J]. Journal of Beijing University of Posts and Telecommunications, 2014, 37 (05): 75-79.)

[64] 李琛，李琦，高军萍，等. 基于 Hadamard 向量的新型 (k+2, k) MSR 码 [J]. 河北工业大学学报，2018, 47 (02): 9-13. (Li Chen, Li Qi, Gao Junping, *et al.* New (k+2, k) MSR codes based on Hadamard vectors [J]. Journal of Hebel University of Technology, 2018, 47 (02): 9-13.)

[65] 马良荔，柳青. 减少重建数据量的冗余编码技术研究 [J]. 计算机科学，2017, 44 (S1): 463-469. (Ma Liangli, Liu Qing. Researches of redundancy coding technologies on reducing reconstruction data amount [J]. Computer Science, 2017, 44 (S1): 463-469.)

[66] 李杰. 面向分布式存储系统的具有最优存取//更新性质的最小存储再生码的设计与分析 [D]. 成都：西南交通大学，2017. (Li Jie. Design and analysis of minimum storage regenerating codes with the optimal access//update property for distributed storage systems [D]. Chengdu: Southwest Jiaotong University, 2017.)

[67] 李晨卉. 应用于分布式存储系统的准循环再生码构造方案 [J]. 计算机工程，2015, 41 (3): 81-87. (Li Chenhui. Construction Scheme of Quasi-cyclic Regenerating Code for Distributed Storage System [J]. Computer Engineering, 2015, 41 (3): 81-87.)

[68] Chen Bin, Xia Shutao, Hao Jie, *et al.* Constructions of optimal cyclic (r, δ) locally repairable codes [J]. IEEE Trans on Information Theory, 2016, PP (99): 1-1.

[69] Park H, Lee D, Moon J. LDPC code design for distributed storage: balancing repair bandwidth, reliability and storage overhead [J]. IEEE Trans on Communications, 2018, PP (99): 1-1.

[70] 肖宜龙. 随机化数据冗余方法及其在存储系统中的应用 [D]. 成都：电子科技大学, 2013. (Xiao Yilong. random data redundancy method and its application in distributed storage systems [D]. Chengdu: University of Electronic Science and Technology of China, 2013.)

[71] 王禹，赵跃龙，侯昉. 基于矩阵运算的最小冗余存储再生码 MSRRC 研究 [J]. 计算机科学，2014, 41 (S2): 191-194+207. (Wang Yu, Zhong Yuelong, Hou Fang. Minimun Redundancy storage regeneration code research msrrc based on matrix operation [J]. Computer Science, 2014, 41 (S2): 191-194+207.)

[72] 谢显中，黄倩，王柳苏，等. 一种云存储中基于干扰对齐的多节点精确修复方法 [J]. 电子学报，2014, 42 (10): 1873-1881. (Xie Xianzhong, Huang Qian, Wang Liusu, *et al.* A multi-node exact repair method in cloud storage based on interference alignment [J]. Acta Electronica Sinica, 2014, 42 (10): 1873-1881.)

[73] 李小兵，许胤龙，林一施，等. X 再生码：一类适用于云存储的准确修复编码 [J]. 计算机应用与软件，2014, 31 (08): 241-244, 248. (Li Xiaobing, Xu Yinlong, Lin Yishi, *et al.* X regenerating codes: a calss of accurate repair codes for cloud storage [J]. Computer Applications and Software, 2014, 31 (08): 241-244+248.)

[74] 王静，张崇，梁伟，等. 分布式存储系统中基于 Pyramid 码的局部性修复编码 [J]. 电子测量与仪器学报, 2017, 31 (9): 1481-1487. (Wang Jing, Zhang Chong, Liang Wei, *et al.* Locally repairable codes based on Pyramid codes in distributed storage systems [J]. Journal of Electronic Measurement and Instrumentation, 2017, 31 (9): 1481-1487.)